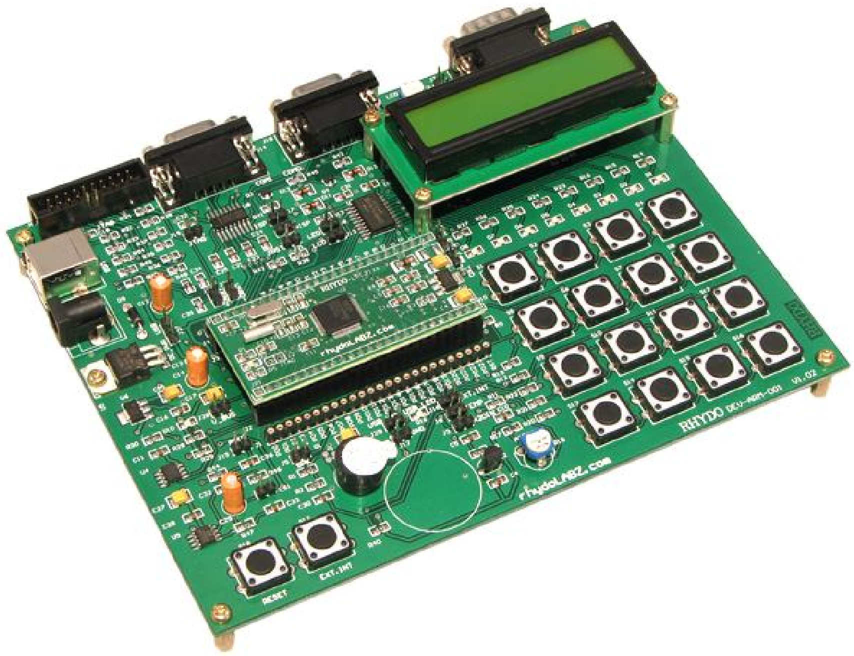
eCee NXP LPC 2129

ARM Development Board

User Manual



Rhydo Technologies (P) Ltd.
(An ISO 9001:2008 Certified R&D Company)
Golden Plaza, Chitoor Road,
Cochin – 682018, Kerala State, India
Phone : 0091- 484-2370444, 2371666
Cell : 0091- 99466 70444
Fax : 0091 - 484-2370579
E-mail : info@rhydolabz.com, sales@rhydolabz.com
WebSite : http://www.rhydolabz.com

rhydolabz.com

CONTENTS

- Copy and Add the CPU Startup Code
- Create New Source Files
- Add Source Files to Project
- Set Tool Options for Target
- Configure Startup Code
- Creating HEX File
- Build Project

❖ Connecting the eCee NXP LPC 2129 Development Board

❖ Programming Steps

- Automatic mode
- Manual mode

❖ JTAG Programming/Debugging

CHAPTER-4 : I/O DISTRIBUTION

❖ The Pin Distribution Of LPC 2129 Development Board

CHAPTER-1: OVERVIEW

The eCee LPC 2129 Development and Evaluation Board from RhydoLabz can be used to evaluate and demonstrate the capabilities of NXP LPC 2129 microcontrollers. The board (with a base board and header board) is designed for general purpose applications and includes a variety of hardware to exercise microcontroller peripherals. The LPC 2129 Board contains all hardware components that are required in a single-chip LPC 2129 system plus 2 COM ports for serial RS232 output, **one CAN port** to test Controller Area Network (CAN) Interface and interfaces like Lcd, Buzzer, Keyboard, Temperature Sensor, Potentiometer, Led's, EEPROM etc.

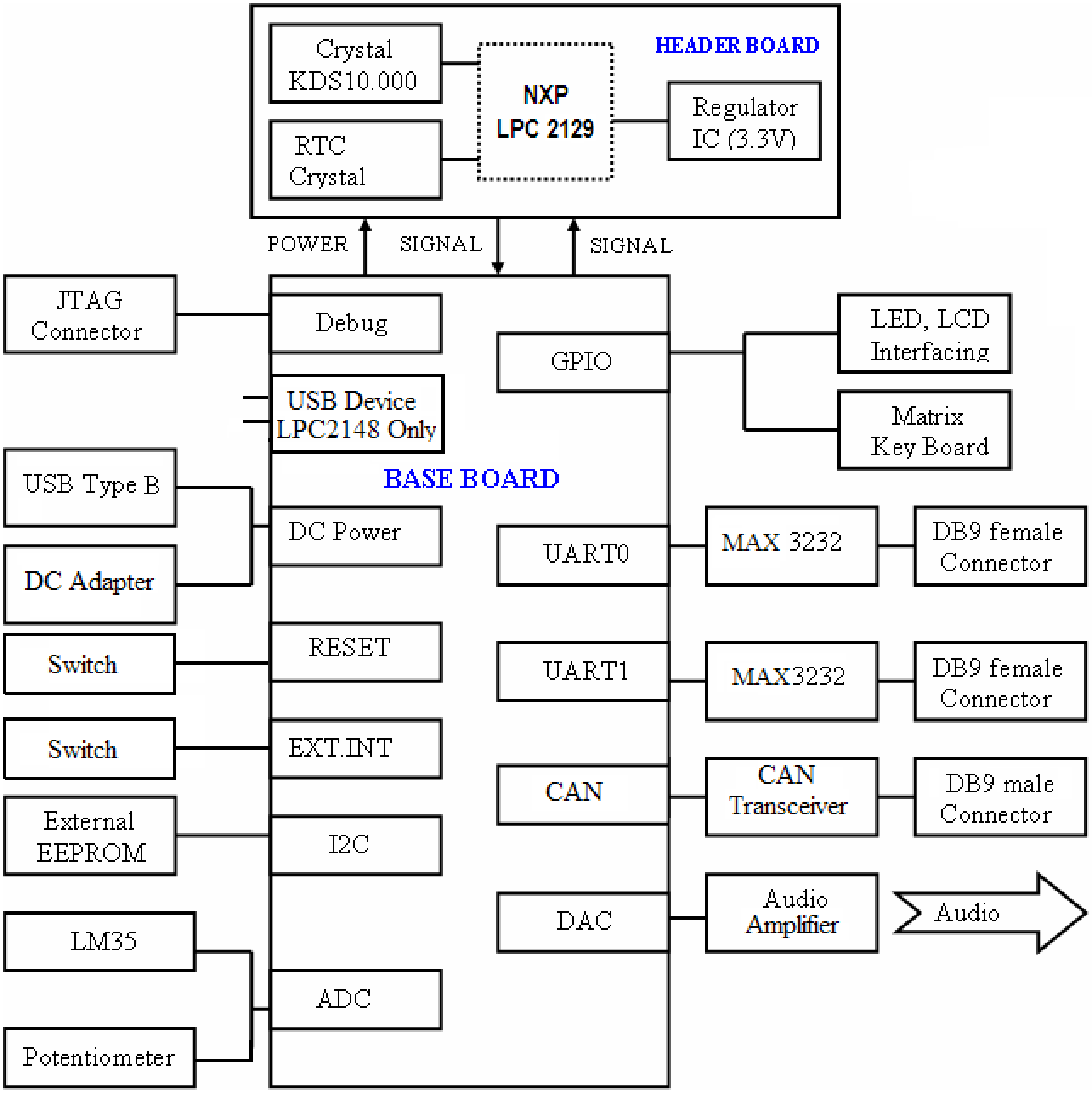
FEATURES

- Includes LPC2129 Header Board with in-built **CAN** peripheral
- No separate programmer required (On-Chip Boot loader)
- No Separate power adapter required (USB power source)
- Two RS-232 Interfaces (For direct connection to PC's Serial port)
- On Board Two Line LCD Display (2x16) (with jumper select option)
- On Board 8 LED Interface to test Port pin (with jumper select option)
- On Board Pot interface to ADC
- On Board Temperature Sensor Interface (Optional)
- On Board Buzzer Interface
- On Board 4x4 (16 Keys) Matrix Keyboard
- On Board I^2C EEPROM
- On Board External Interrupt Button
- On Board Connector for PWM Output
- PWM controlled LCD backlight
- On Board Connector for Analog Output
- On Board Speaker Output
- LF Amplifier LM 386
- Adaptor (any standard 9-12V power supply) option

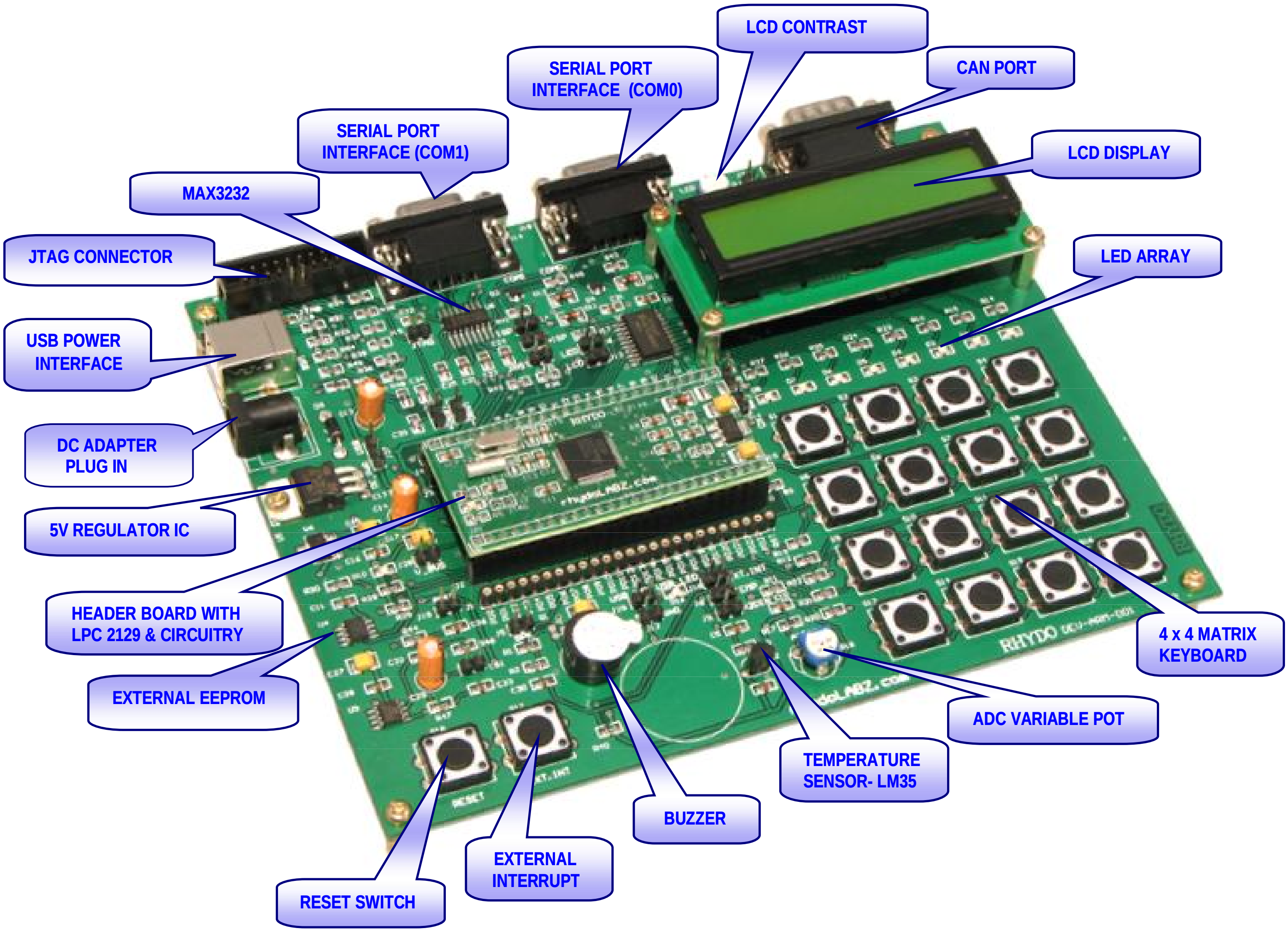
- On Board Power LED Indicator
- On Board Reset button
- All Port Pins available at Berge Strip
- On Board JTAG Connector for Debugging/Programming
- Power Supply Reverse Polarity Protection
- Controller Area Network (CAN) transceiver
- Three On Board DB9 Connectors (Two for UART and One for **CAN**)
- On Board USB Connector
- On Board 1 Amp Voltage Regulator
- On Board Connector for regulated 3V3 output
- On Board Connector for regulated 5V output
- Can be used as main board for developing applications

CHAPTER-2: HARDWARE INTRODUCTION

FUNCTION MODULE



INTERFACE OVERVIEW



PERIPHERALS	DESCRIPTION
CON1	USB Jack
CON2	Power Jack
JTAG (JP1)	JTAG Connector
COM0	UART0 Interface via Female DB9 Connector
COM1	UART1 Interface via Female DB9 Connector
LCD CONT.	LCD Contrast Adjust through Pot
CAN PORT	CAN Interfacing through Male DB9
LCD1	LCD Display Module, HD44780U
D1- D8	Light Emitting Diodes
S1- S16	Matrix Keypad Switches
EXT.INT (S17)	External Interrupt Key
RESET (S18)	Reset Key
A2DIN (R18)	Potentiometer as ADC input
U1	HDP1206 X (Buzzer)
U2	LM35 (Temperature Sensor)
U3	74LV244 (LED Driver IC)
U4	External EEPROM 24C04
U5	LM7805 (Regulator IC- 5V)
U6	LD1117S33 (Regulator IC- 3.3V)
U7	74LV244 (LCD Driver IC)
U8	MAX3232 (Level Converter)
U9	LM386 (Audio Amplifier)
U10	TJA1040 (CAN interfacing IC)

JUMPER LIST

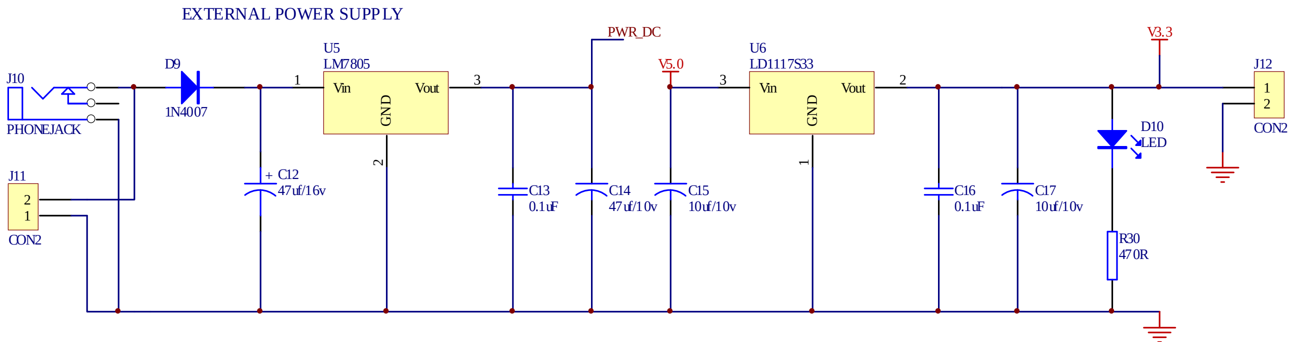
Jumper no:	Description	Set option	Set description
J1	Microcontroller Pin-outs for External Access	Already Set in Package	Already Set in Package
J2	Header Board Connector Pins	Already Set	Already Set in Package
J3	Header Board Connector Pins	Already Set	Already Set in Package
J4	Microcontroller Pin-outs for External Access	Already Set	Already Set in Package
J5	BUZ	Short to access	Buzzer function activate
J6	TEMP	Short to access	LM35 Access via AD0.0
J7	LED PWM	1-2 2-3	LCD Brightness vary on PWM LCD works normally
J8	LED	Short to access	LED enable (P1.16-P1.23)
J9	A2DIN	Short to access	Potentiometer as AD0.1 input
J11	CON2	Short to access	Choose DC Power Plug In
J12	+3V3-GND	As Required	Provide a 3.3V to an External Module
J13	LCD	Short to access	LCD Module Functions
J15	JTAG	Short to access	JTAG Access
J16	EXT.INT	Short to access	External Interrupt Input Access
J17	ISP	Short to access	On Program Download
J19	AOUT	Short to access	Analog Out
J20	ISP*	Short to access	Board is RESET Externally
J21	JRST	Short to access	Short while Program Downloading
J22	USBD -	Short to access	USB Device Access
J23	USBD+	Short to access	USB Device Access
J24	CANRXD	Short to access	CAN Data Reception (LPC2129 Only)
J26	CAN TXD	Short to access	CAN Transmission (LPC2129 Only)
J27	Supply Select	1-2 2-3	DC Power is selected, short J11 USB Power is selected
J28	V_BUS	Short to access	USB Device Access
J29	USB LNK	Short to access	USB Led Functioning

POWER SUPPLY

eCee NXP LPC 2129 Board has two power supplies; you can choose one of the following ways to supply power

(1) Through an Adaptor (any standard 9-12V power supply)
(2) Through the motherboard USB port

The external Power Supply circuit is given below:



CLOCK SOURCE

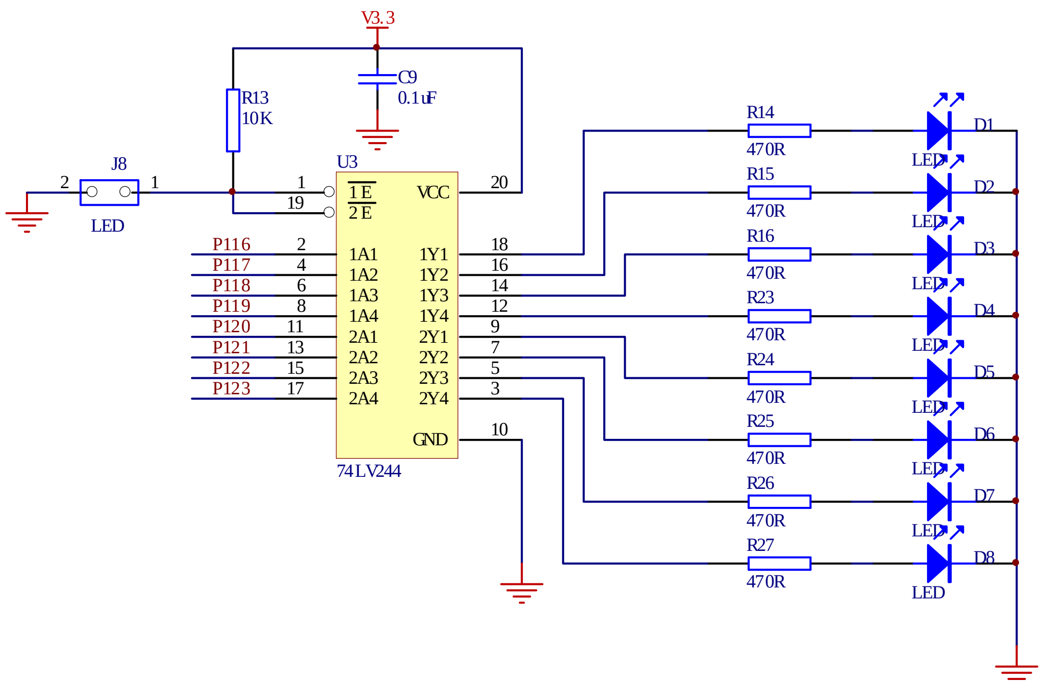
eCee NXP LPC 2129 evaluation board has two clock sources:

- 32.768 KHz Crystal as the RTC clock source
- 10 MHz Crystal as the MCU clock source

LED INTERFACING

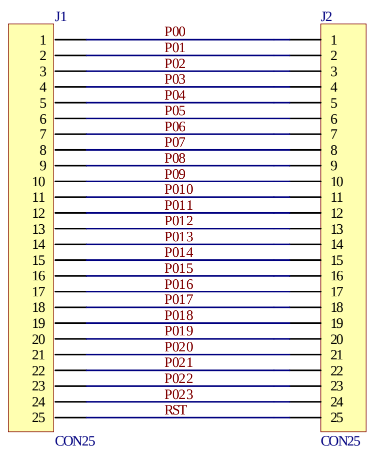
LED’s are the simplest devices to test port functioning. The board contains 8 LED’s connected to PORT1 pins 16 to 23 (P1.16 to P1.23).

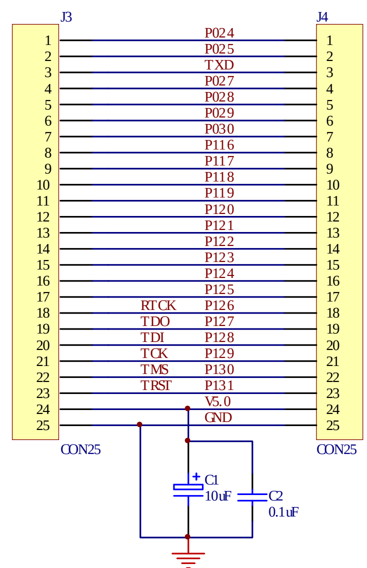
*📝**Note:** For the LED’s to work, the LED jumper (J8) must be in position.*



PORT PINS – BERGE STRIP & ROUND MACHINE CUT CONNECTOR

The board has all port pins available at Berge strip and at round machine cut female connector. The connection is as given below.

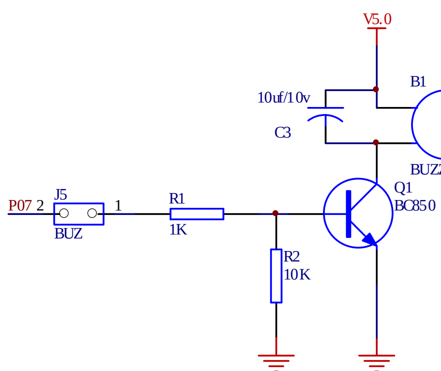


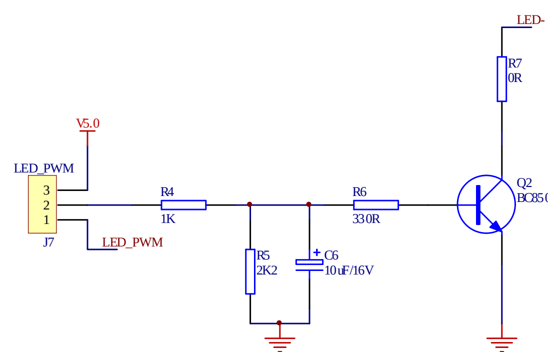


BUZZER INTERFACE & LED-PWM

Buzzer is connected to P0.7

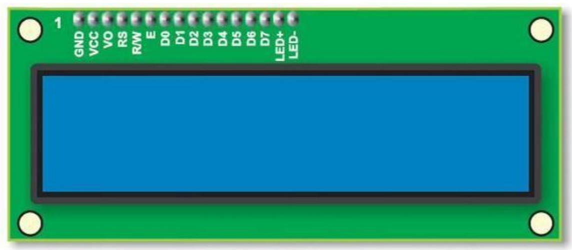
Note: *For the Buzzer to work, the jumper (J5-BUZ) must be in position.*





LCD - LIQUID CRYSTAL DISPLAY

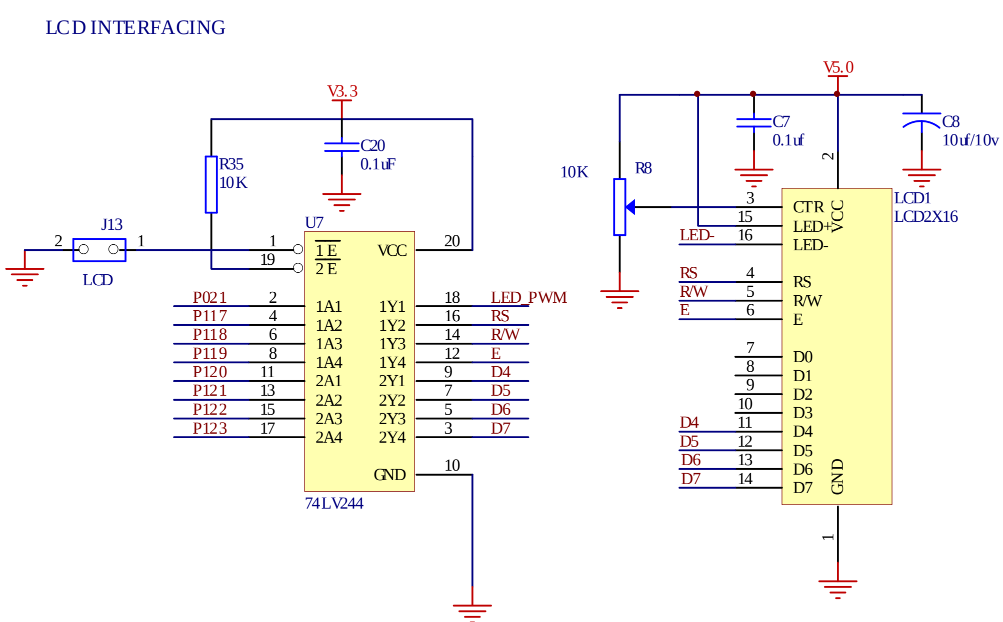
The display is a standard 2x16 LCD which displays 2 lines of 16 characters. Each character is 40 pixels, making it 1280 pixels overall. The display receives ASCII codes for each character at the data inputs (D0–D7).



Connection Diagram

The LCD module can be used in 4-bit or 8-bit mode. The module uses HD44780U (from Hitachi) as the controller IC. The eCee LPC 2129 development board **uses 4-bit interface**. PORT1 pins (P1.17 – P1.23) are used for data/command control pins. An On-Board potentiometer enables to adjust the LCD contrast to a better view in every angle.

Note: For the LCD module to work, the LCD jumper (J13) must be in position



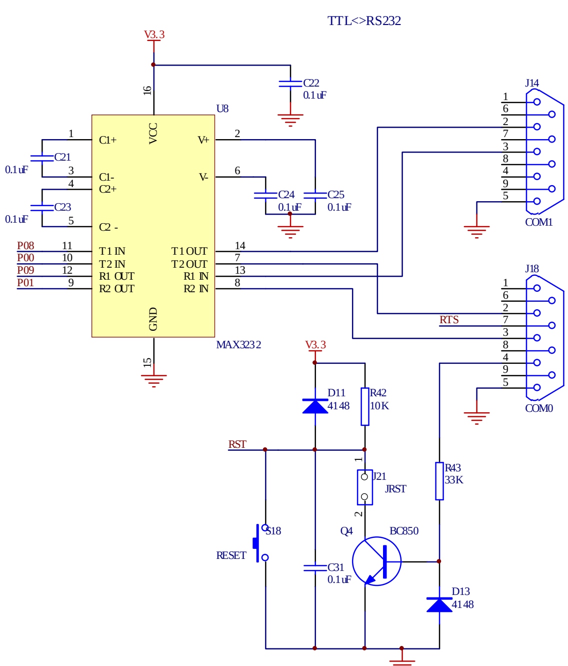
The Lcd module has 16 pins for interfacing out of which 8 are data pins (DB0-DB7) and 3 (RS,RW,EN) are control pins.

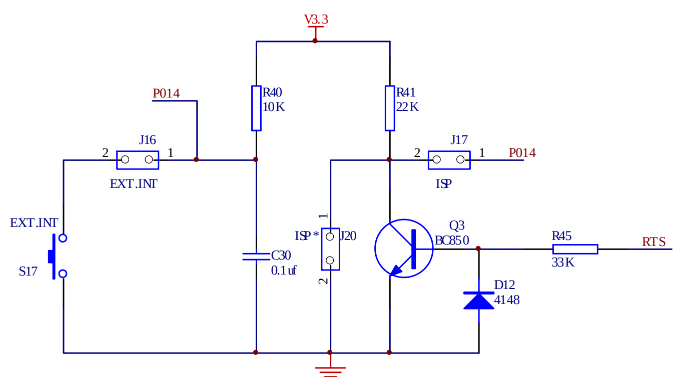
Pin No:	Functional Description	I/O Port
1	VSS	GND
2	VCC	+5V
3	VEE	CONTRAST
4	RS	P1.17
5	R/W	P1.18
6	E	P1.19
7	DB0	NC
8	DB1	NC
9	DB2	NC
10	DB3	NC
11	DB4	P1.20
12	DB5	P1.21
13	DB6	P1.22
14	DB7	P1.23
15	LED+	+5V
16	LED-	GND

UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

The LPC 2129 microcontrollers comes with two UART modules - UART0 and UART1. It has only asynchronous (no clock connection) mode of operation. The UART0 operates through **P0.0 (TXD)** and **P0.1 (RXD)** pins while the UART1 uses **P0.8 (TXD)** and **P0.9 (RXD)** pins. The LPC 2129 UART output itself operates at CMOS voltages, and needs an external serial line driver to convert its output into a higher symmetrical line voltage. The MAX 3232 serial driver serves this purpose.

Note:* For *the UART module to work, the ISP (J17) and RST (J21) jumpers must be removed.

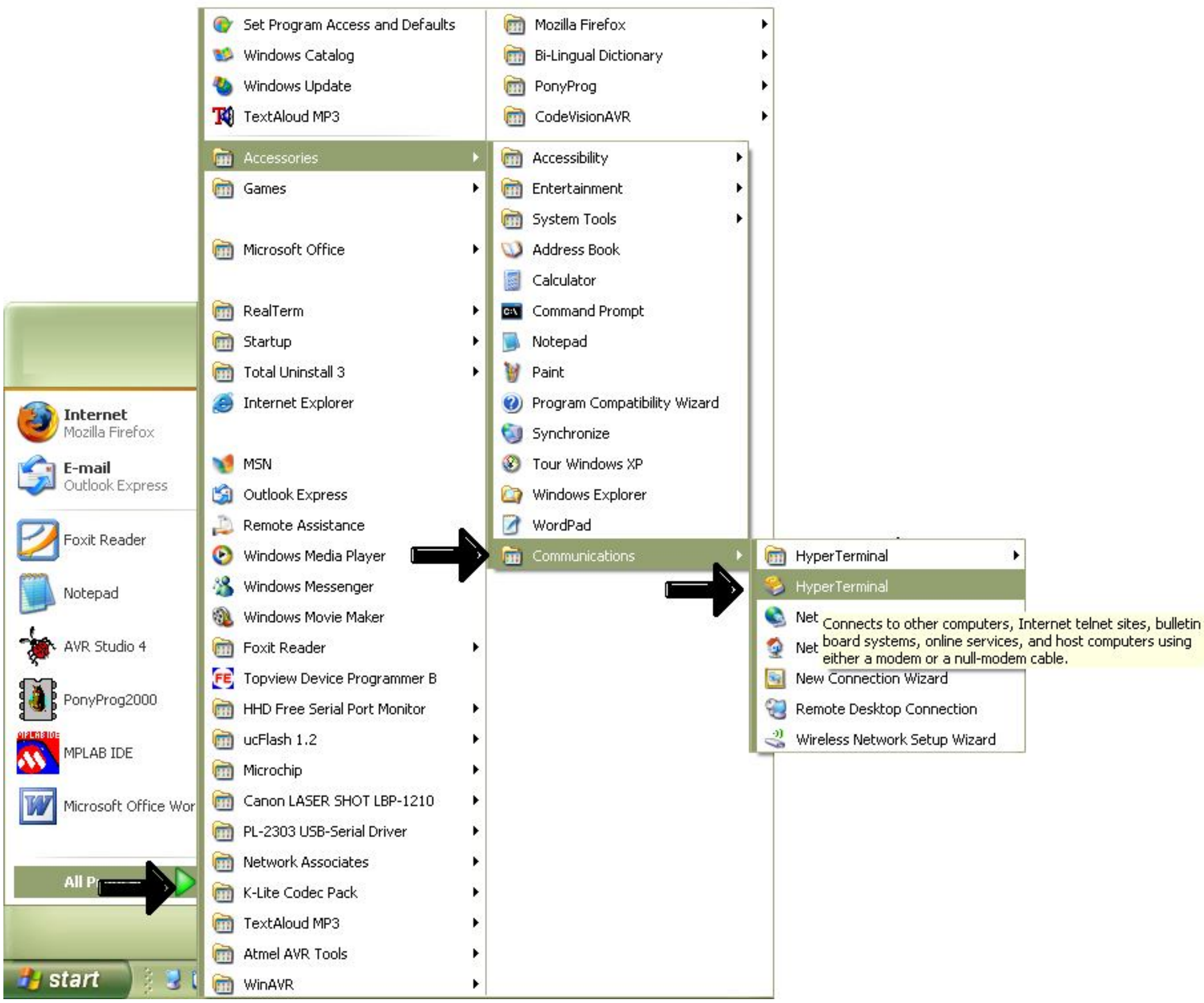




CREATING HYPERTERMINAL IN PC

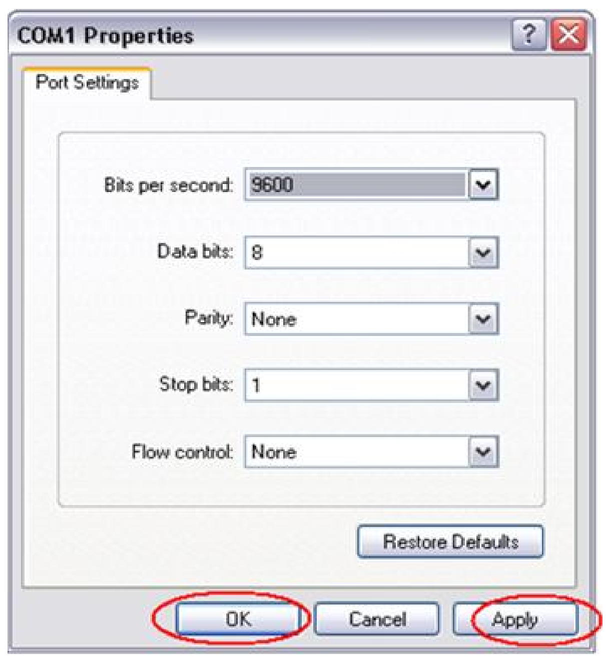
The serial data transmitted through UART can be viewed on a PC using a Windows tool for Serial Port Communication called HyperTerminal.

Steps To Create Hyper Terminal

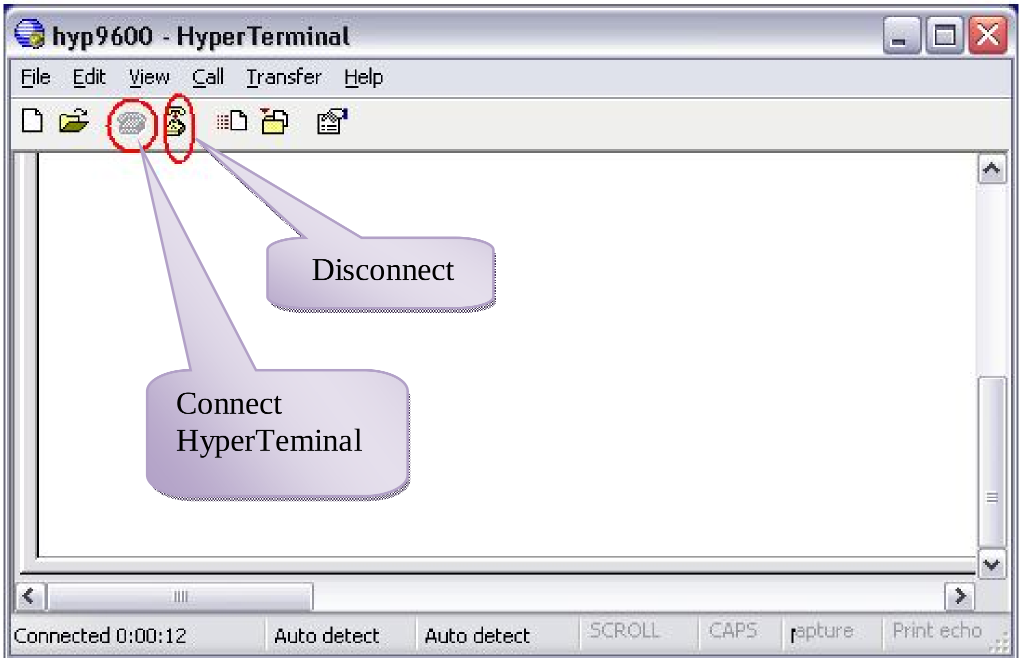


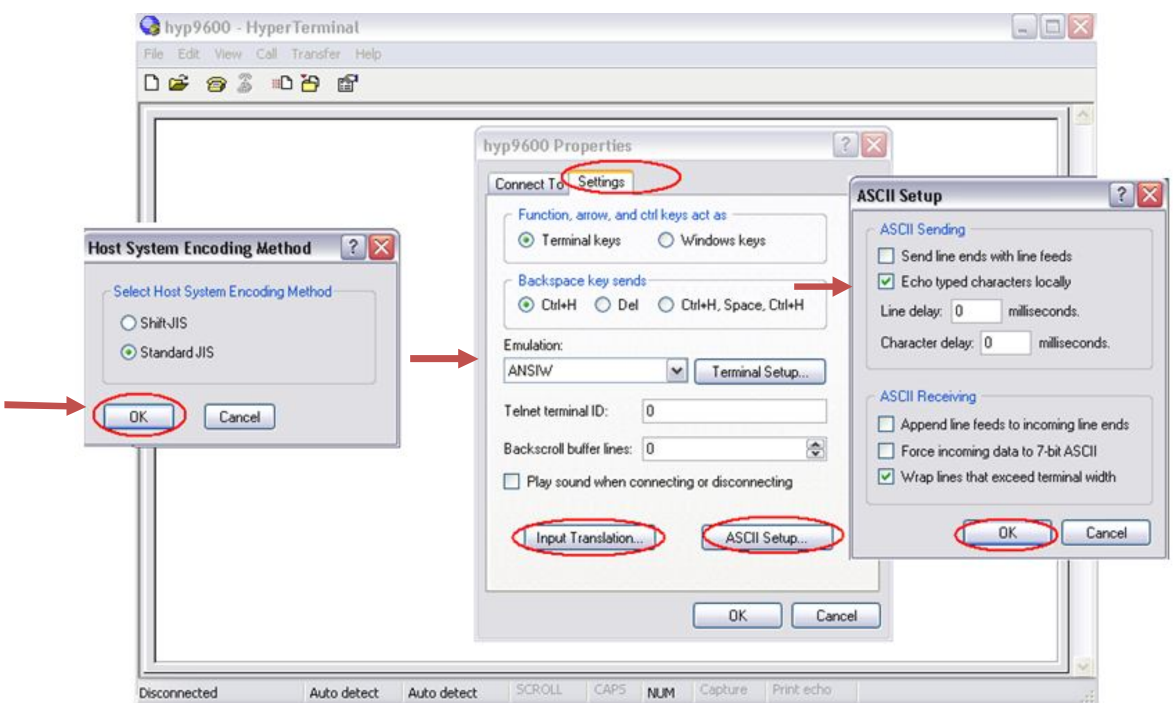




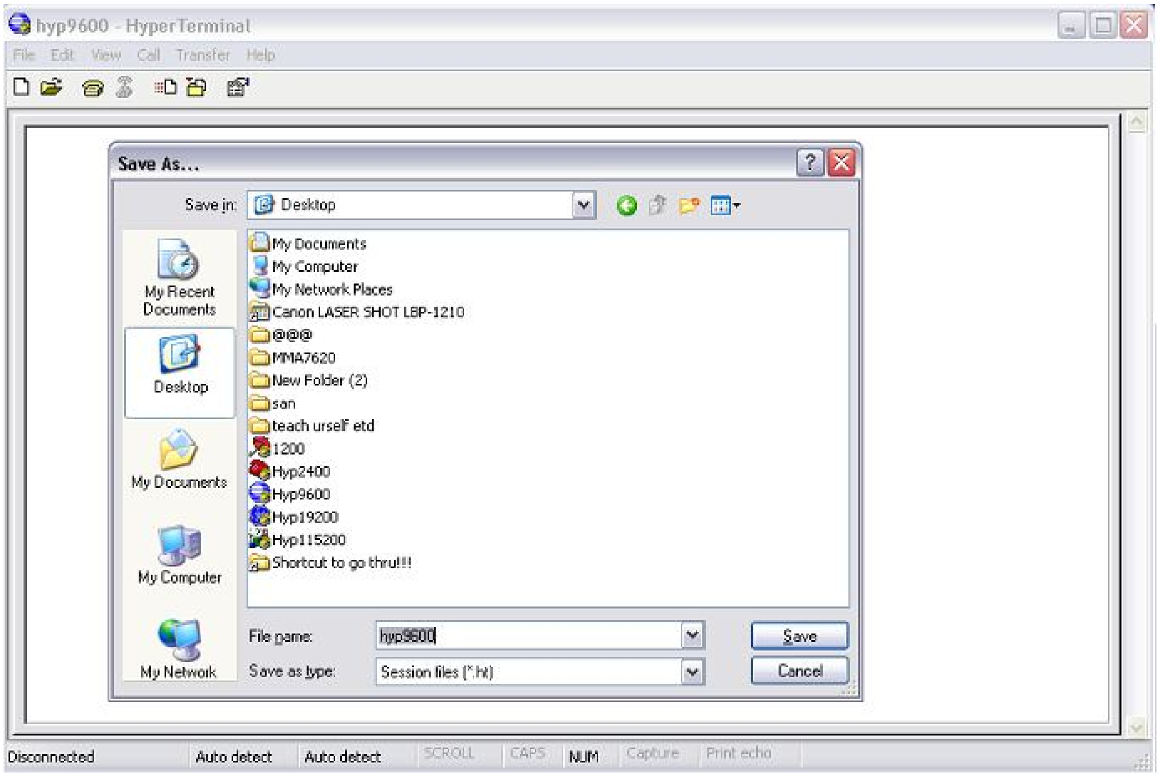


Now Disconnect the Hyper Terminal. Then open the Property from 'File'.


hyp9600 - HyperTerminal
File Edit View Call Transfer Help
Disconnect
Connect
HyperTeminal
Connected 0:00:12
Auto detect
Auto detect
SCROLL
CAPS
Capture
Print echo

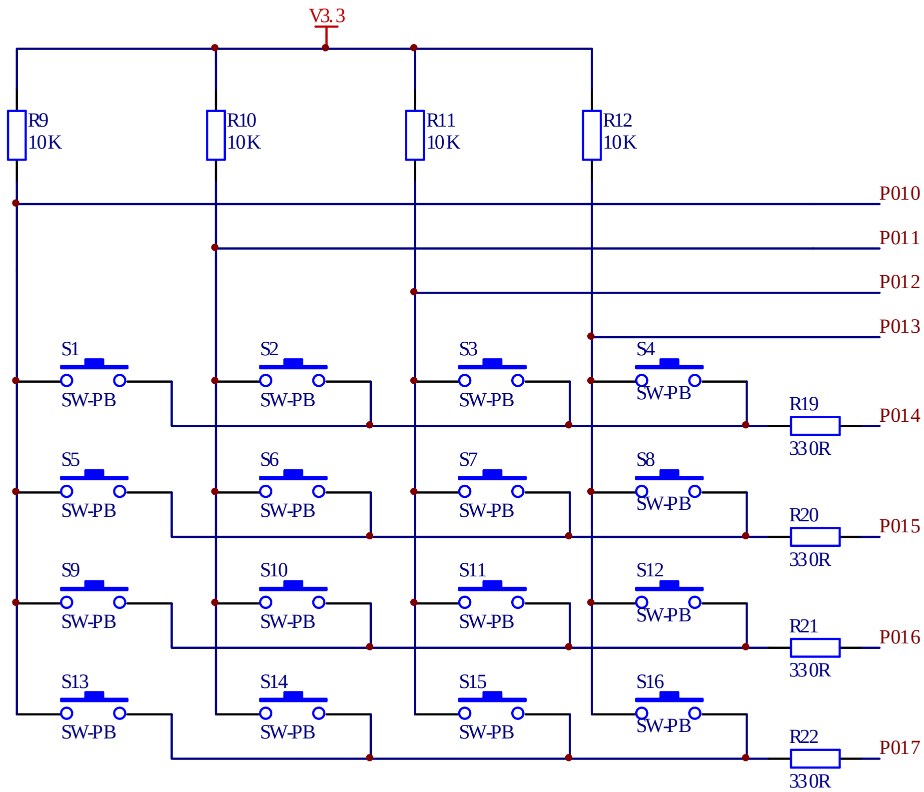

hyp9600 - HyperTerminal
File Edit View Call Transfer Help
hyp9600 Properties
Connect To
Settings
Function, arrow, and ctrl keys act as
Terminal keys
Windows keys
Backspace key sends
Ctrl+H
Del
Ctrl+H, Space, Ctrl+H
Emulation:
ANSIW
Terminal Setup...
Telnet terminal ID:
Backscroll buffer lines:
Play sound when connecting or disconnecting
Input Translation...
ASCII Setup...
OK
Cancel
Host System Encoding Method
Select Host System Encoding Method
Shift-JIS
Standard JIS
OK
Cancel
ASCII Setup
ASCII Sending
Send line ends with line feeds
Echo typed characters locally
Line delay: 0 milliseconds.
Character delay: 0 milliseconds.
ASCII Receiving
Append line feeds to incoming line ends
Force incoming data to 7-bit ASCII
Wrap lines that exceed terminal width
OK
Cancel
Disconnected
Auto detect
Auto detect
SCROLL
CAPS
NUM
Capture
Print echo

Select File → Save As → Desktop



MATRIX KEYBOARD

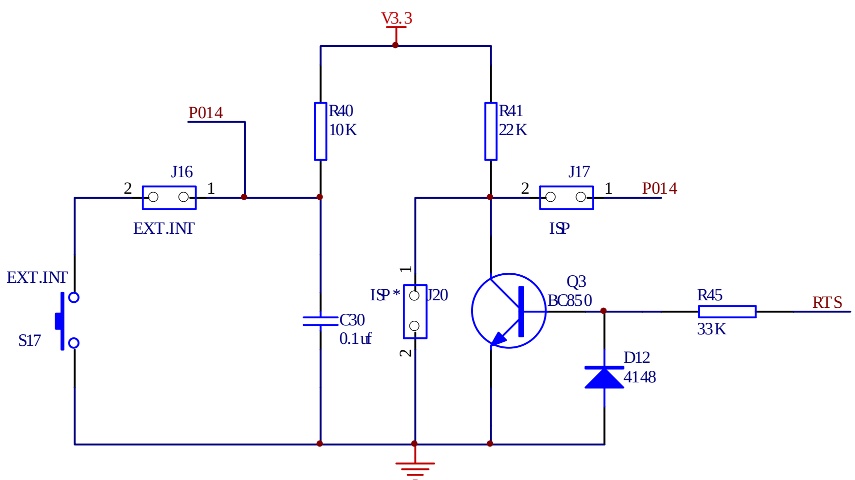
A keypad is simply an array of push buttons connected in rows and columns, so that each can be tested for closure with the minimum number of connections. There are 16 keys on a phone type pad arranged in a 4X4 matrix. Assume the columns are labeled 1, 2, 3, 4 and the rows A, B, C, D. If we assume that all the rows and columns are initially high, a keystroke can be detected by setting each row low in turn and checking each column for a zero. In the KEYPAD circuit the 8 keypad pins are connected to Port0. While coding, **Pins P0.14-P0.17 should be initialized as outputs and pins P0.10-P0.13 as inputs**. These input pins are pulled high to logic 1. The output rows are also initially set to 1. If a 0 is now output on row A, there is no effect on the inputs unless a button in row A is pressed. If these are checked in turn for a 0, a button in this row which is pressed can be identified as a specific combination of output and input bits.



EXTERNAL INTERRUPT

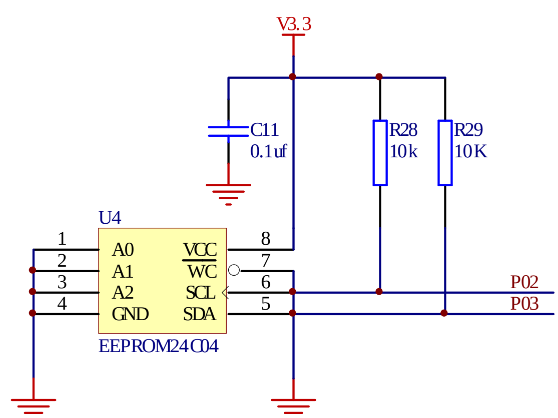
This development board uses External Interrupt 1 - EINT1 (Button S17). It is connected to P0.14.

Note: Place jumper on EXT_INT (J16) pin for the proper functioning of external interrupt.

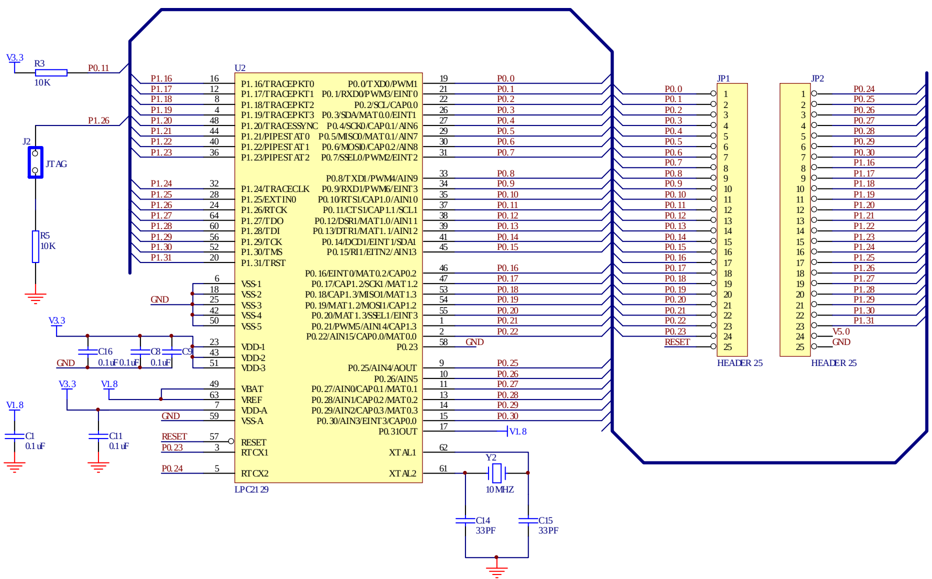


EXTERNAL EEPROM INTERFACING

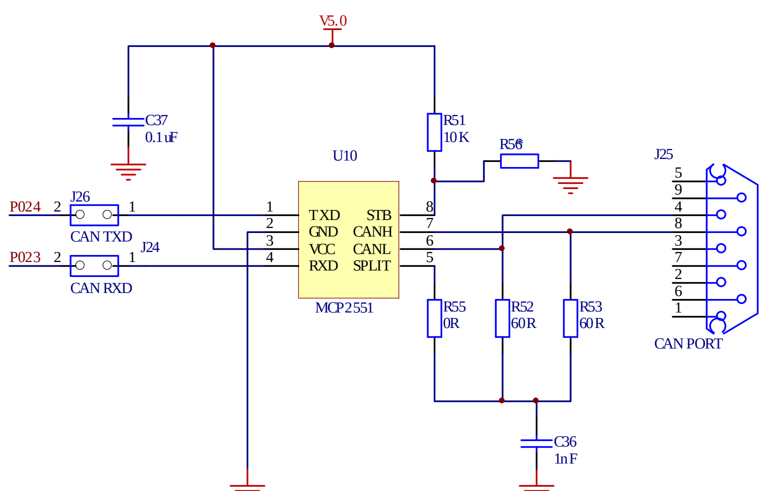
The eCee LPC 2129 development board has 4K SERIAL EEPROM interfacing. Internally organized with 256 pages of 2-byte each, the 4K requires an 9-bit data word address for random word addressing with data transfer rate 100 kbits/s. Also it is to be noted that an external EEPROM (AT24C04) is the slave device to be communicated with the microcontroller, via I^2C protocol.



PIN OUT DIAGRAM



CONTROLLER AREA NETWORK (CAN) INTERFACE

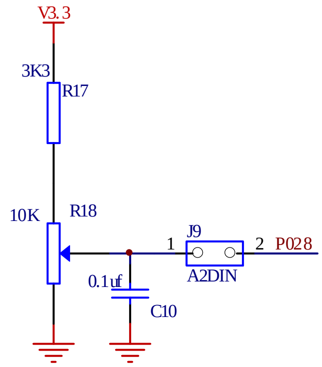


ADC POTENTIOMETER

The eCee LPC 2129 development board has a potentiometer connected to its ADC pin **P0.28**.

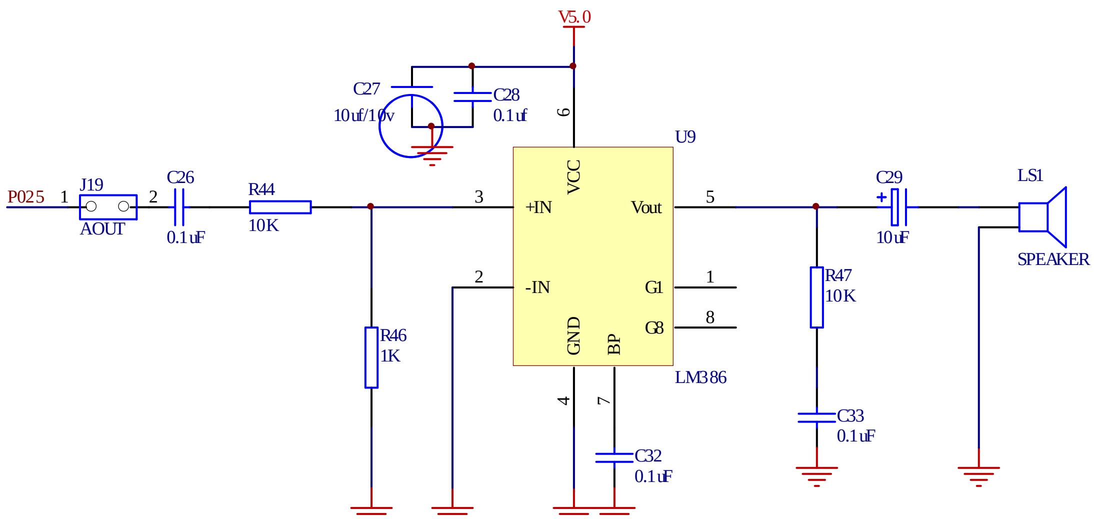
Note: Place jumper on A2DIN (J9) pin .

ADC PORT



AUDIO AMPLIFIER

AUDIO AMPLIFIER



CHAPTER-3 : SOFTWARE DEVELOPMENT

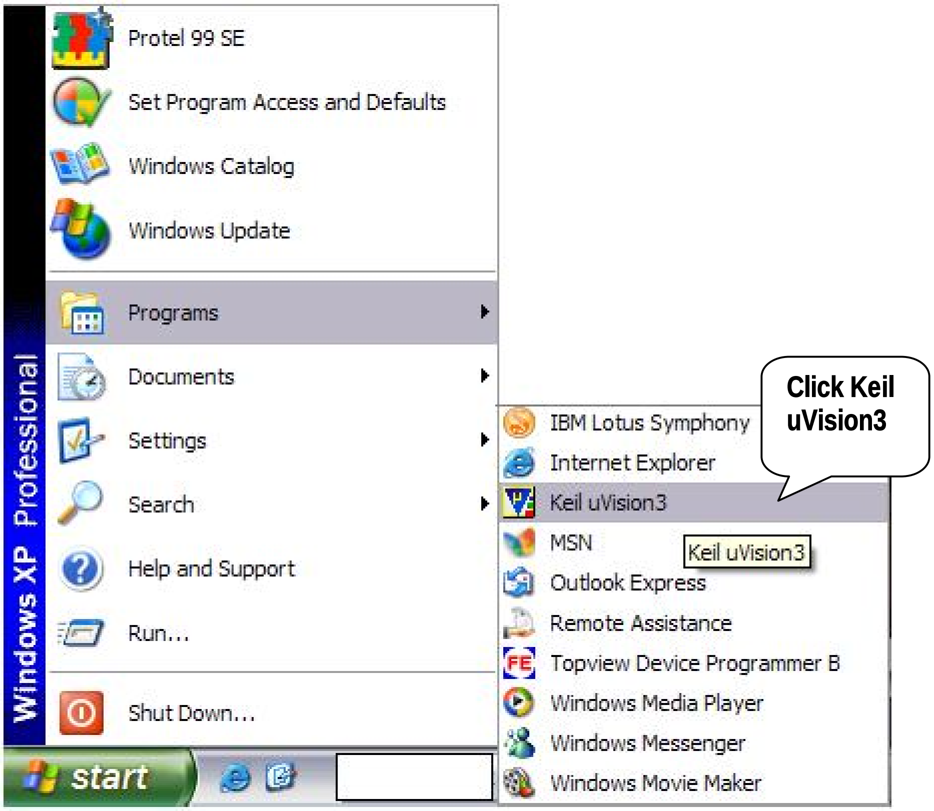
FAMILIARIZATION OF KEIL PROFESSIONAL DEVELOPMENT SUITE

CREATING A PROJECT

Keil µVision3 software is a standard Windows application for project development and can be installed like any standard application. µVision3 includes a project manager which makes it easy to design applications for an ARM based microcontroller. You need to perform the following steps to create a new project:

- Open ***Kiel uVision3*** Software from ***start*** menu or Desktop shortcut
- Select the Toolset (only required for ARM Projects).
- Create Project File and Select CPU.
- Create New Source Files.
- Add Source Files to the Project.
- Set Tool Options for Target Hardware.
- Configure the CPU Startup Code.
- Create a HEX File.
- Build Project and Generate Application Program Code.

The section provides a step-by-step tutorial that shows you how to create a simple µVision3 project.



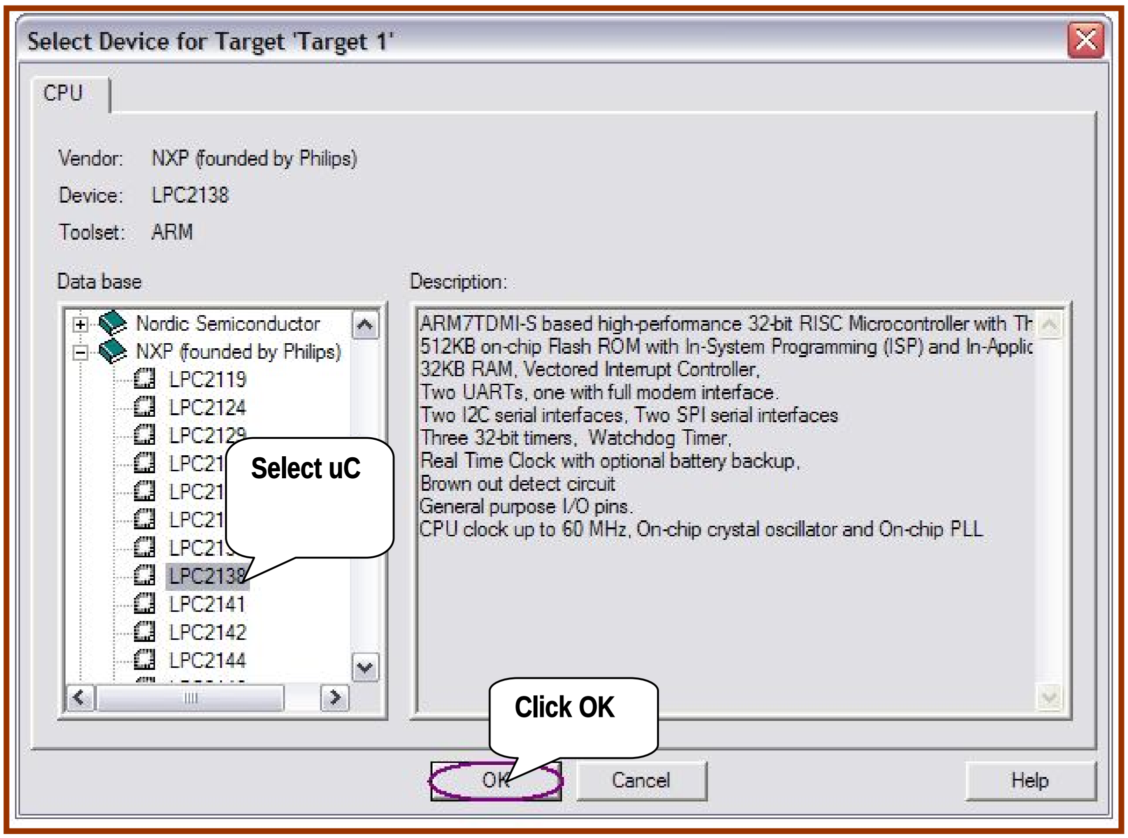
Creating Project File

To create a new project file select from the µVision3 menu **File – New – µVision Project….** This opens a standard Windows dialog that asks you for the new project file name. You should use a separate folder for each project. You can simply use the icon **Create New Folder** in this dialog to get an empty new folder.

Select this folder and enter the file name for the new project, eg. **Project1**. µVision3 creates a new project file with the name **PROJECT1.UV2** which contains a default target and file group name. You can see these names in the **Project Workspace – Files**.

Select Microcontroller from Device Database

When you create a new project µVision3 asks you to select a CPU for your project. The **Select Device** dialog box shows the µVision3 device database. Just select the microcontroller you use. For the example in this chapter we are using the Philips LPC 2138 controller. This selection sets necessary tool options for the LPC2138 device and simplifies in this way the tool configuration.



Copy and Add the CPU Startup Code

An embedded program requires CPU initialization code that needs to match the configuration of your hardware design. This Startup Code depends also on the tool chain that you are using. Since you might need to modify that file to match your target hardware, the file should be copied to your project folder.

For most devices, µVision3 asks you to copy the CPU specific Startup Code to your project. This is **required on almost all projects** (exceptions are library projects and add-on projects). The Startup Code performs configuration of the microcontroller device and initialization of the compiler run-time system. Answer with **YES** to this question.



Note: The CPU Startup Code typically requires some configuration; however the default configuration gives you a good starting point for single chip applications.

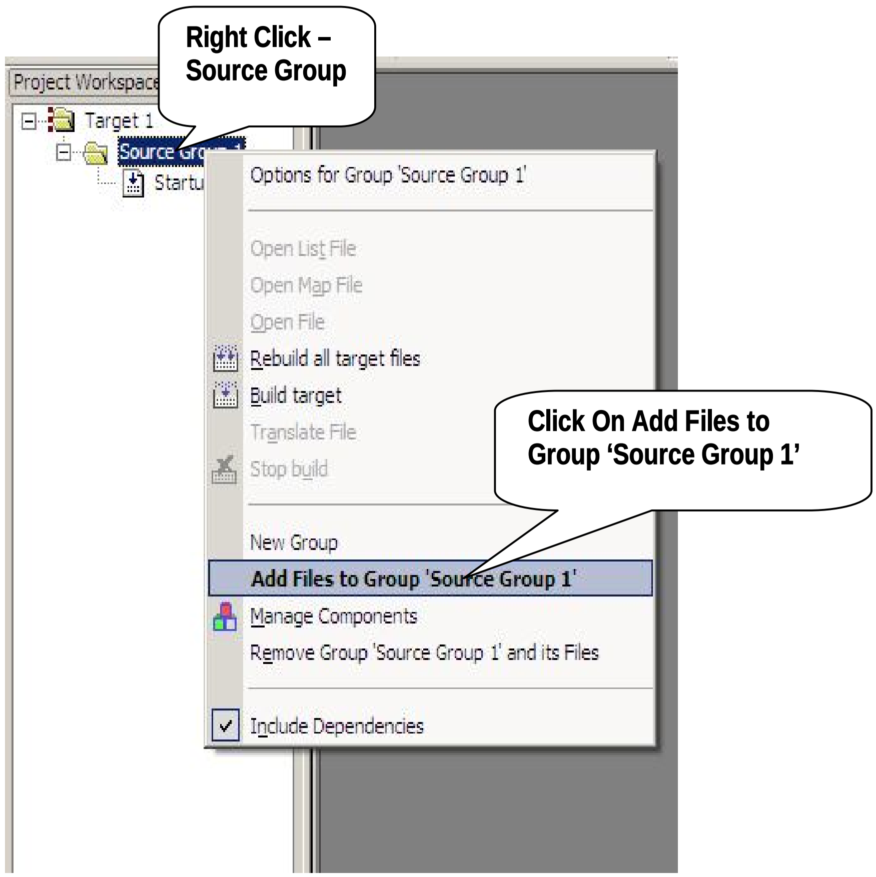
Create New Source Files

You may create a new source file with the menu option **File – New**. This opens an empty editor window where you can enter your source code. µVision3 enables the C color syntax highlighting when you save your file with the dialog **File – Save As…** under a filename with the extension *.C. Here we save our example file under the name **led.C**.

```
#include <LPC21xx.h>                /* LPC 2129 definitions          */
void delay(void) {                  /*  Delay function               */
int i;
for(i=0;i<1000;i++);                /* Delay for LED blink           */
 }
int main (void)  {
 IO1DIR = 0x00FF0000;               /* P1.16..23 defined as Outputs */
 while (1) {                        /* infinite Loop                 */
 IO1SET = 0X00FF0000;               /*  P1.16..23 pins high          */
 delay();
IO1CLR = 0x00FF0000;                /* P1.16..23 pins low            */
delay();  } }
```

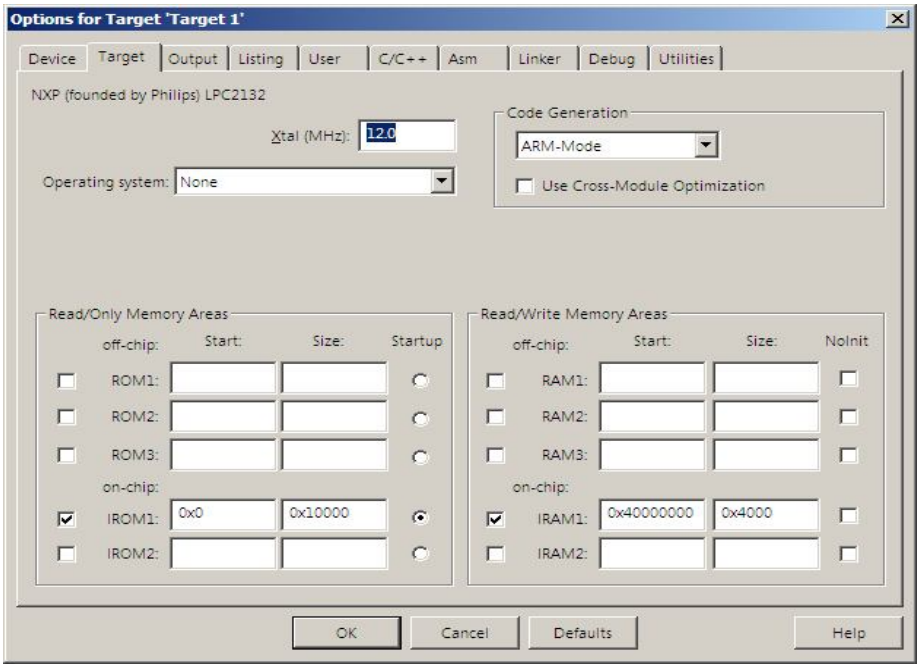
Add Source Files to Project

Once you have created your source file you can add this file to your project. µVision3 offers several ways to add source files to a project. For example, you can select the file group in the **Project Workspace – Files** page and click with the right mouse key to open a local menu. The option **Add Files** opens the standard files dialog. Select the file **led.c** you have just created.



Set Tool Options for Target

µVision3 lets you set options for your target hardware. The dialog **Options for Target** opens via the toolbar icon or via the **Project - Options for Target** menu item. In the **Target** tab you specify all relevant parameters of your target hardware and the on-chip components of the device you have selected. The following dialog shows the settings for our example.

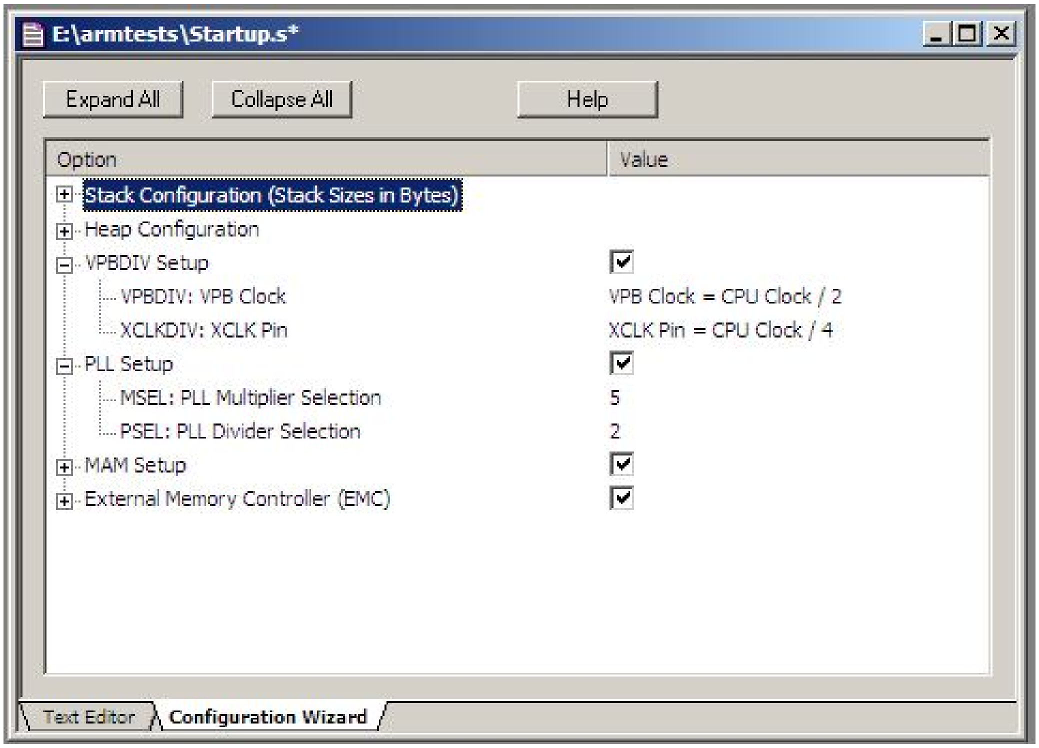


The following table describes the options of the **Target** dialog:

Dialog Item	Description
Xtal	Specifies the external clock frequency of your device. Several microcontrollers use an on-chip PLL to generate the CPU clock. In this case the value is not identical with the XTAL frequency. Check your hardware design carefully to determine the correct value.
Operating System	Allows you to select a Real-Time Operating System for your project.
Use On-chip ROM / RAM	Defines the address spaces for the on-chip memory components for the linker/locater. Note that on some devices you need to reflect this configuration in the Startup Code.

Configure Startup Code

The CPU Startup Code (on most ARM targets the file name is **Startup.S**) may be open from the **Project Workspace - Files Tab**. Most startup files have embedded comments for the µVision3 Configuration Wizard which provides menu driven selections.

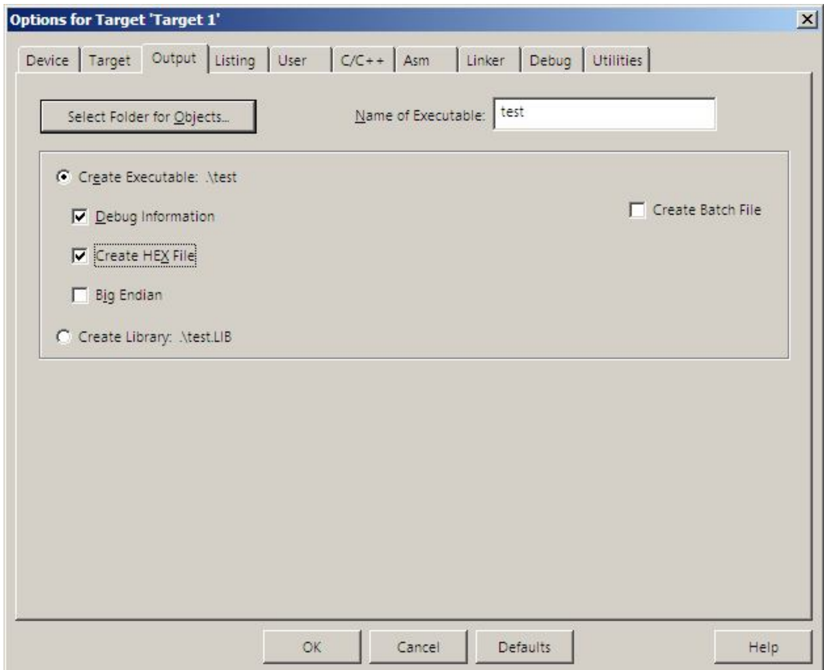


The default settings of the Startup Code give a good starting point on most single chip applications. However you need to adapt the configuration for your target hardware. CPU/PLL clock and BUS system are target specific and cannot be automatically configured. **You need to ensure** that the settings in the startup file match the other settings in your project.

The button **Edit as Text** opens the Startup Code in a standard editor window and allows you to review the source code of this file.

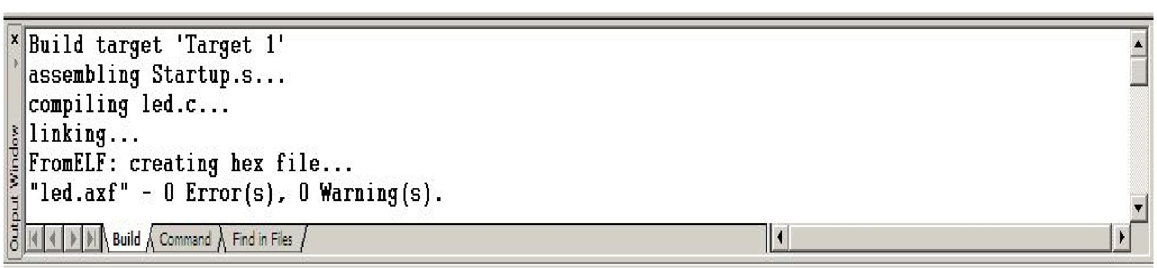
Create HEX File

Once you have successfully generated your application you can start debugging. After you have tested your application, it is required to create an Intel HEX file to download the software into an EPROM programmer or simulator. µVision3 creates HEX files with each build process when **Create HEX file** under **Options for Target – Output** is enabled.

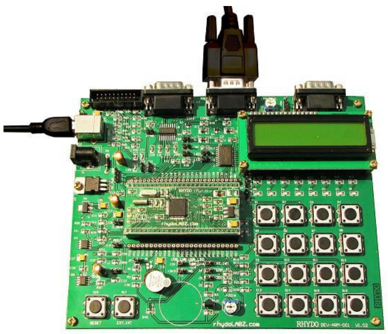


Build Project

Typical, the tool settings under **Options – Target** are all you need to start a new application. You may translate all source files and link the application with a click on the **Build Target** toolbar icon. When you build an application with syntax errors, µVision3 will display errors and warning messages in the **Output Window – Build** page. A double click on a message line opens the source file on the correct location in a µVision3 editor window.

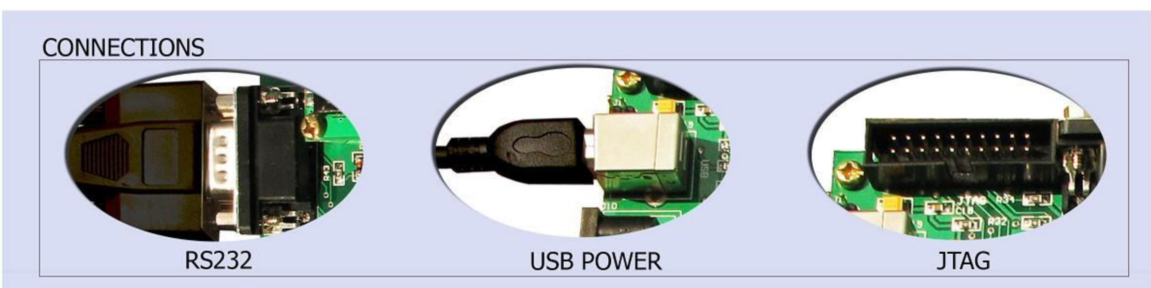


CONNECTING THE DEVELOPMENT BOARD



The eCee LPC 2129 requires a +9-12V DC adapter or USB connection for power and either a serial connection for In-system programming, or the JTAG connector.
Use LPC2000 Flash Utility Software at the PC side for programming through Serial port. Connect eCee LPC 2129 Development board to your PC using USB cable (for powering it) and serial cable (for In-serial programming) as shown in the figure.

***Note**: Serial cable should be **connected to COM0** Port of the development board for downloading. **Corresponding jumpers** has to be inserted (see circuit diagram) for the functioning of each peripheral.*

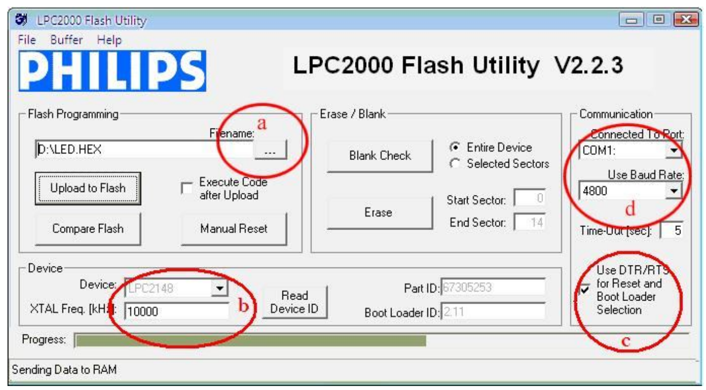


PROGRAMMING STEPS

eCee NXP LPC 2129 development board supports **two modes** **of programming**

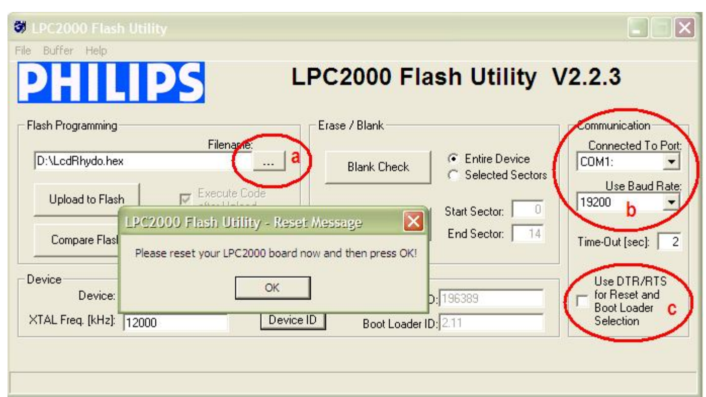
I. AUTOMATIC MODE

1. Configure LPC Flash Utility software at the PC side
a. Browse your hex file here.
b. Select Crystal frequency here.
c. Enable DTR/RTS for Reset and boot loader selection.
d. Select your COM port and Set baud rate (9600) here.
2. Connect system serial port to COM port of eCee LPC 2129.
3. ***Put jumper*** *on* ***RST(J21) , ISP(J17)*** *and Reset the board*
4. Click "Read Device ID" and wait till Device Id is shown
5. Click "Upload to Flash" button in the flash utility software and wait till the programming is over.
6. Remove jumper on ISP and RST
7. Now Reset the development board



II. MANUAL MODE

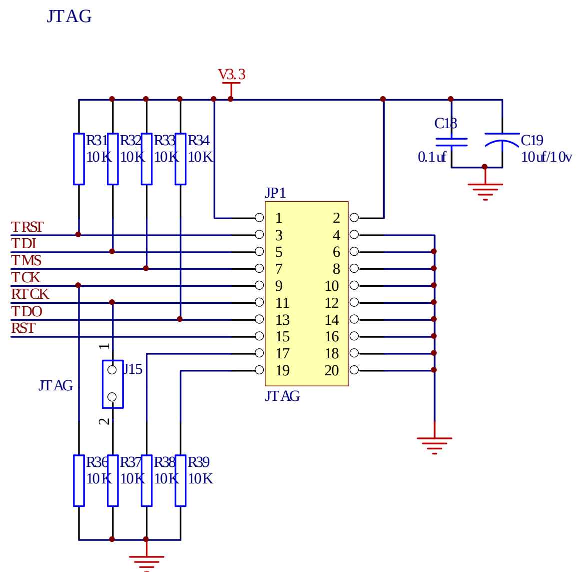
1. Configure LPC Flash Utility software at the PC side
a. Browse your hex file here.
b. Select your COM port and Set baud rate (9600) here.
c. Disable DTR/RTS for Reset and boot loader selection.
2. Connect system serial port to COM port of eCee LPC 2129.
3. ***Put jumper*** *on* ***ISP(J17),*ISP(J20)***
4. Click "Read Device ID"
5. The ***software prompts you*** *to reset the development board.*
6. Press Reset button (S18) and press OK
7. Wait till Device Id is shown
5. Click "Upload to Flash" button in the flash utility software and wait till the programming is over.
*6. Remove jumper on ISP (J17),*ISP (J20)*
7. Now Reset the development board



JTAG INTERFACE FOR PROGRAMMING/DEBUGGING

(The JTAG explained in this section is the ARM-JTAG Debugger/Programmer from Rhydolabz.com)

Note : Connect the JTAG debugger to the development board and make sure that the JTAG Jumper in the development board (Marked JTAG – J15) is inserted.



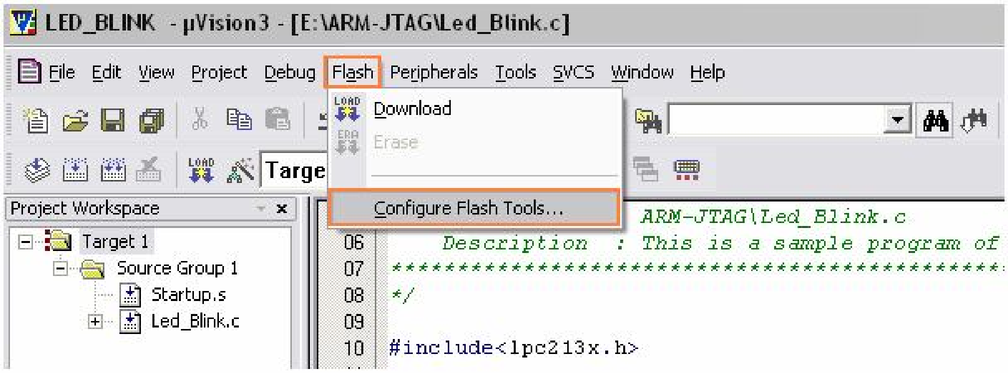
This section explains how to interface **ARM-JTAG**, a wiggler compatible module, with **ARM evaluation board**. This is a stable and easy to use device which provides a low cost alternative for programming/debugging a target board. It supports Remote Debug Interfacing (RDI).

This **ARM-JTAG** module is compatible with **H-JTAG** software which is free software available for H-JTAG debugging process.

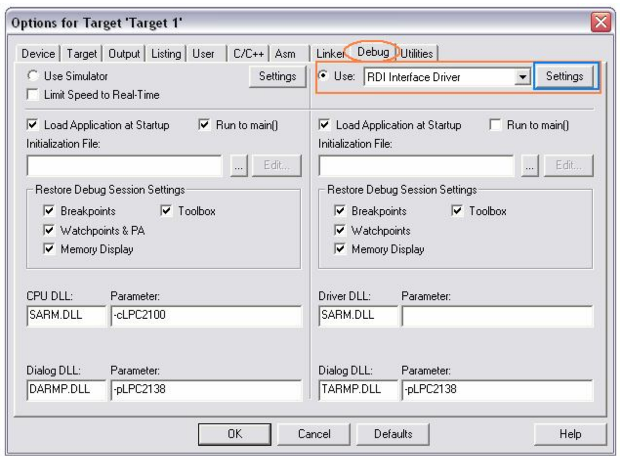
Install the H-JTAG software in your PC.

The example screens showed here uses **uVision 3V3.50 from Keil** Software.

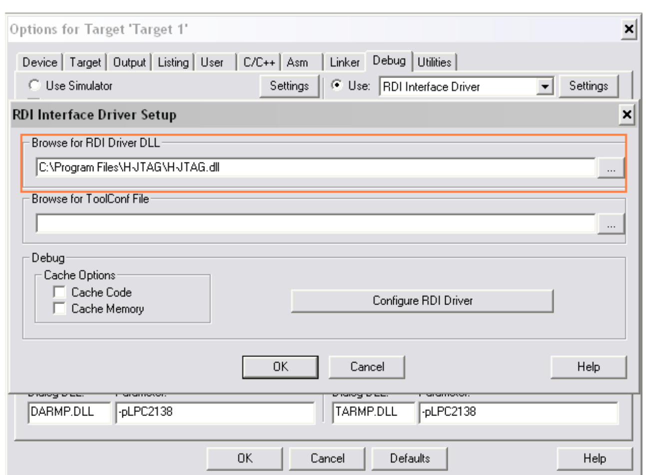
Before you start downloading and debugging process, certain settings have to be configured in Kiel uV Project. First, create a new **Kiel uV project** or open an already existing one, add the needed program in source group and build it. Make sure that the **.hex** file is created. Now select **Flash** menu and click **Configure Flash Tools** in it.



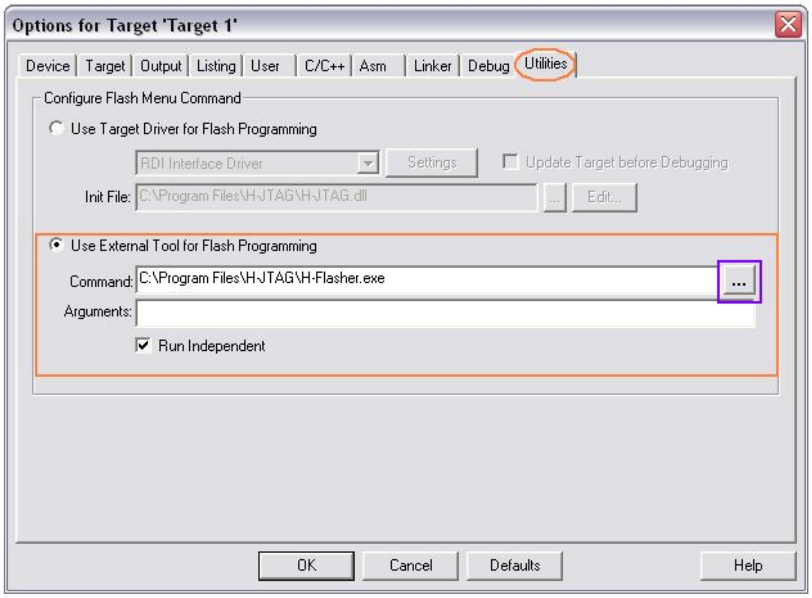
Select **RDI Interface Driver** and click on **Settings** in the **Debug** option. Check whether **Load Application at Startup** and **Run to main** are enabled.



Click **Browse** in the **"Browse for RDI Driver DLL"** option and select the **H-JTAG.dll** file from the installed HFlasher location. (Usually found in C:\Program Files\H-JTAG\H-JTAG.dll). Click "**OK**".



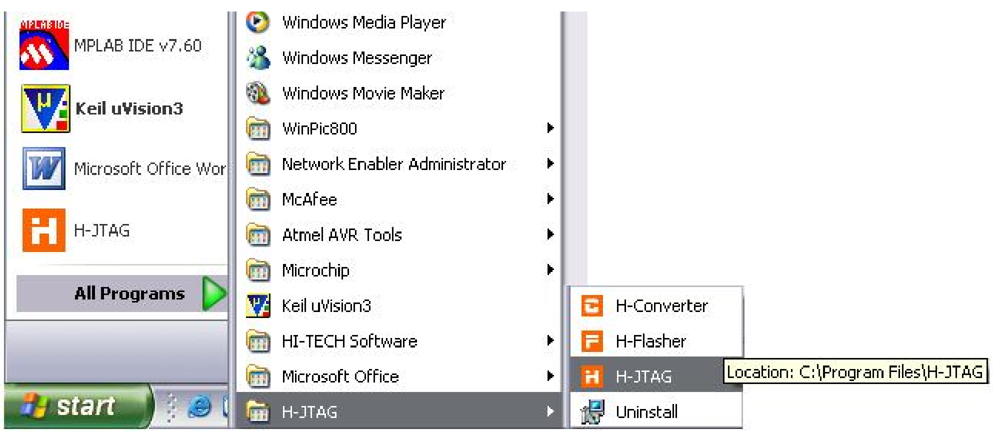
Click **Utilities** enable **Use External Tool for Flash Programming. Click Browse** in the **Command Option** and select the **file H-Flasher.exe** (Usually found in C:\Program Files\H-JTAG\H-Flasher.exe). Enable **Run Independent** option. Click OK. This will exit you from the Configuration Menu.



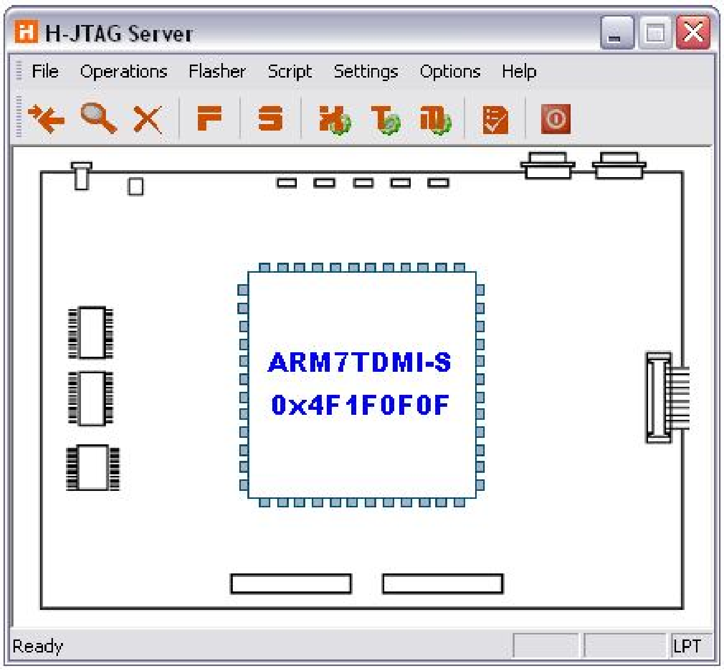
Debugging/Programming

Connect your evaluation board to the JTAG debugger. Make sure that **the JTAG is connected to the parallel port of your PC** using the cable provided.

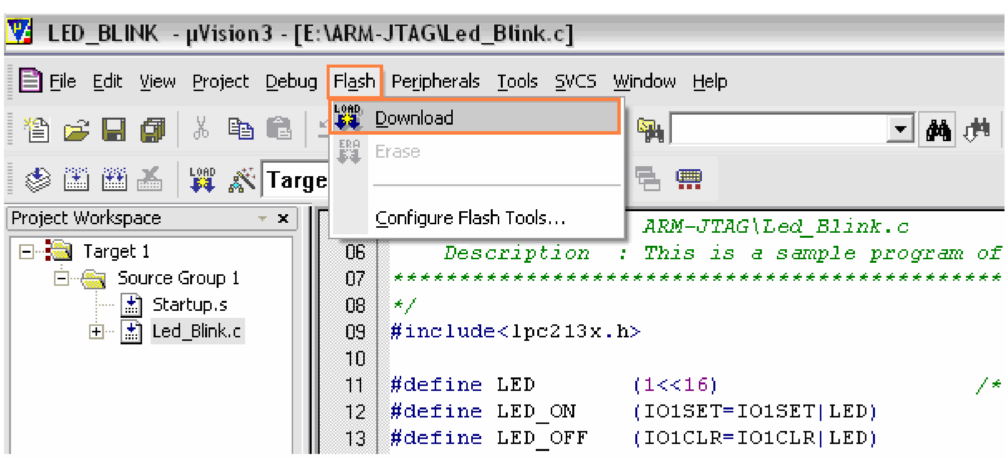
Launch **H-JTAG** software from ***start*** menu.



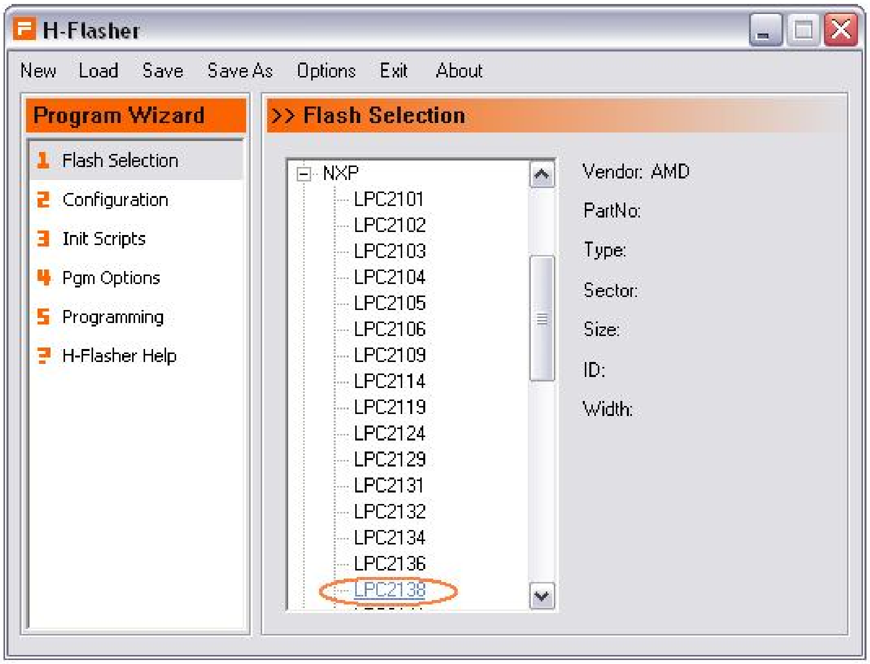
It detects device id of the controller connected to the evaluation board. If the device is not detected, RESET the evaluation board and Click “Detect Target”(In Operations Menu) again. In the example, we use **LPC 2138** microcontroller from NXP with **ARM7TDMI-S** processor.



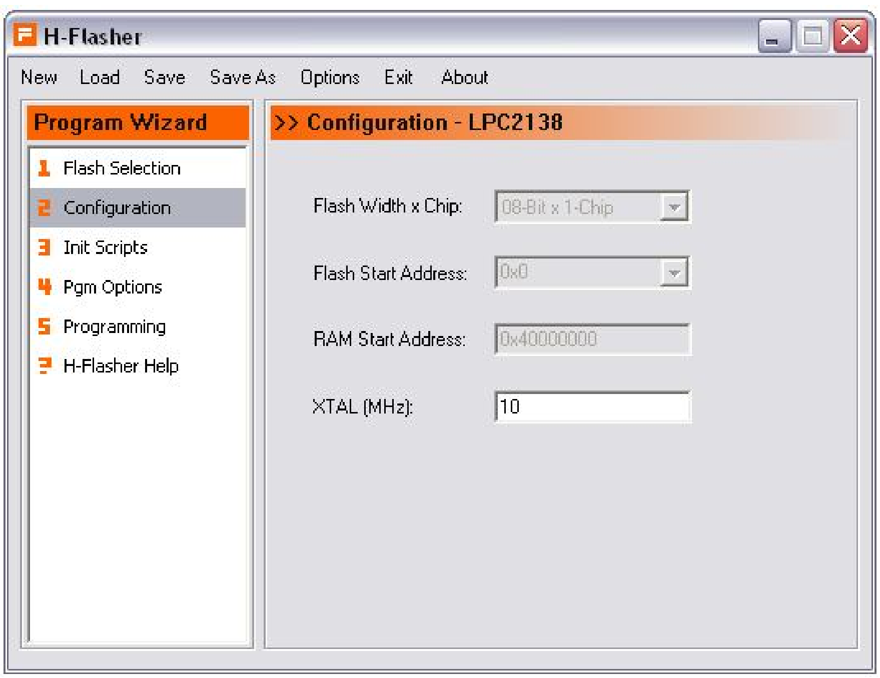
Minimize the H-JTAG screen and pop-up the Keil window. Click **Download** in the Flash Menu for downloading the program using JTAG. The H-JTAG automatically pop-up.



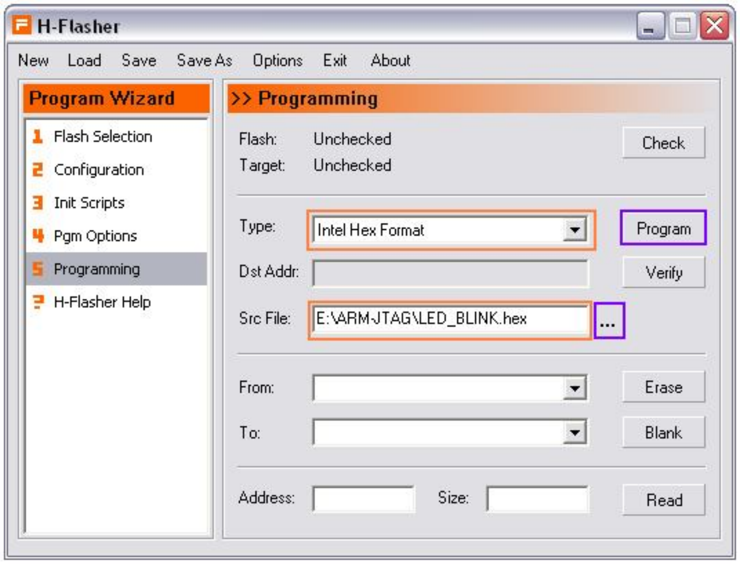
Select **Flash Selection** under **Program Wizard** to select the microcontroller in the evaluation board. Here we use **NXP's LPC2138** controller.



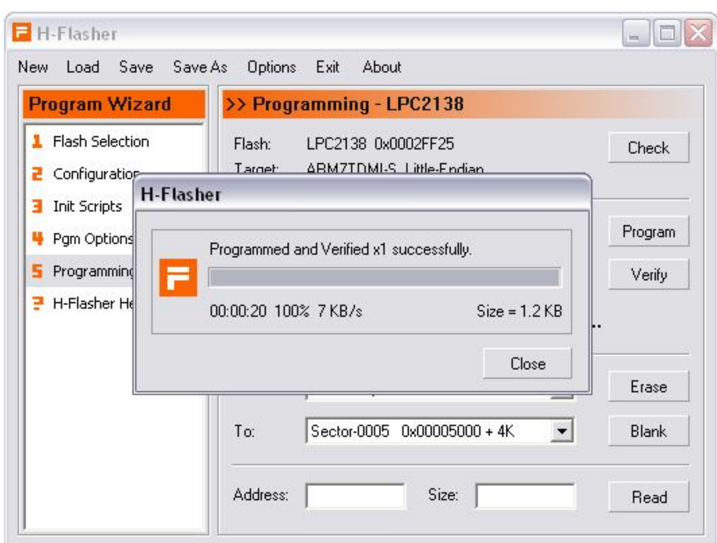
Now select **Configuration** and set external crystal frequency in the **XTAL (MHz)** text box. Enter the frequency of the crystal used in the evaluation board.



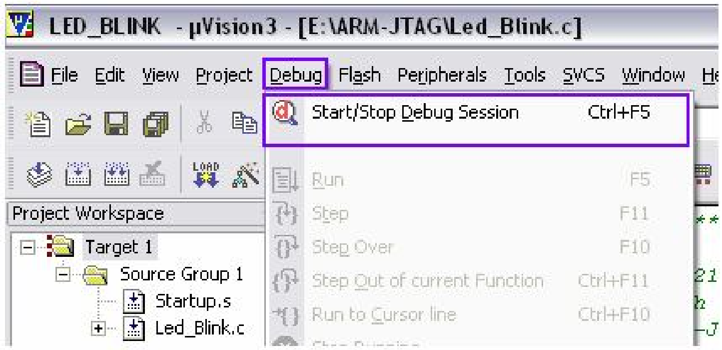
Now Select the **Programming** option and set **Type** as **Intel hex Format** and select the **Source File** from your program folder. We can also check the **Flash** and **Target** unit by clicking **Check.** Now click **Program** button to program the Target board.

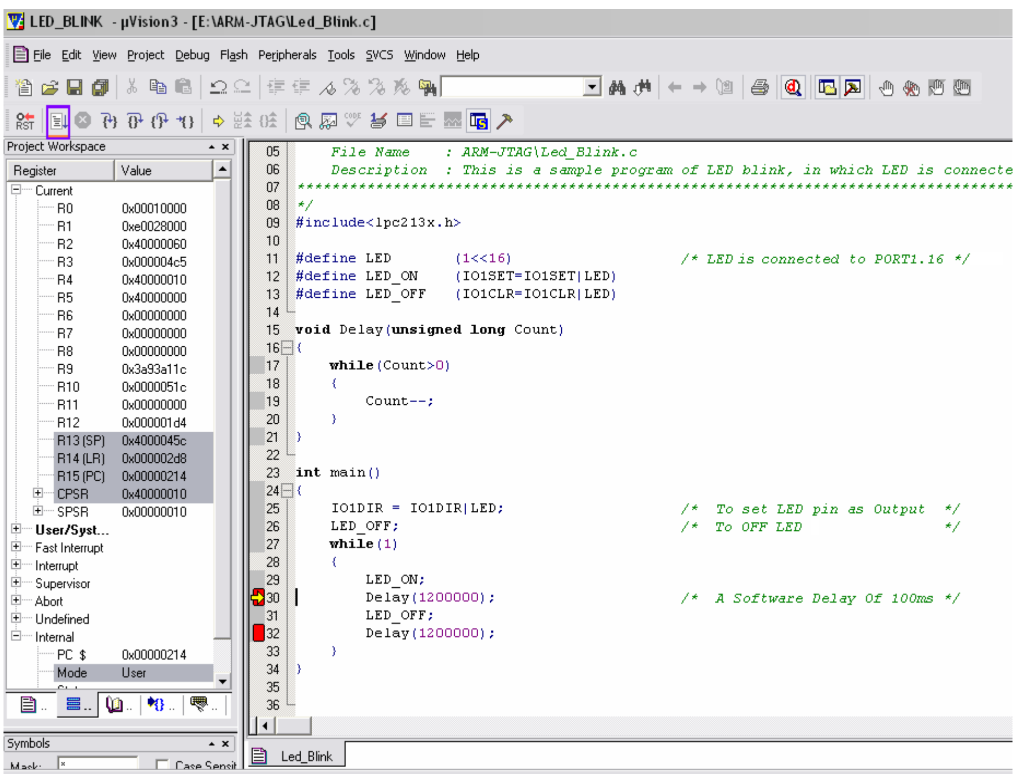


Successful programming is shown as given below, indicating program size.



Now select **Start/Stop Debug Session** from **Debug** menu in the Keil Software.





CHAPTER- 4: I/O DISTRIBUTION

The Pin Distribution of LPC2129 Development Board

LPC2129 Pin No:	Name	Type	The I/O assign of LPC2129 Development Board
1	P0.21	I/O	PWM5
2	P0.22	I/O	N/C
3	RTXC1		
4	P1.19	I/O	Led4/LCD (E/D)
5	RTXC2		
6	VSS		GND
7	VDDA		+3V3
8	P1.18	I/O	Led3/LCD (R/W)
9	P0.25	I/O	AOUT
10	P0.26	I/O	USBD+
11	P0.27	I/O	USBD-
12	P1.17	I/O	Led2/LCD (RS)
13	P0.28	I/O	AD0.1
14	P0.29	I/O	N/C
15	P0.30	I/O	N/C
16	P1.16	I/O	Led1
17	P0.31	I/O	USBLINK
18	VSS		GND
19	P0.0	I/O	TXD0
20	P1.31	I/O	TRST
21	P0.1	I/O	RXD0
22	P0.2	I/O	SCL0
23	VDD		+3V3
24	P1.26	I/O	RTCK

25	VSS		GND
26	P0.3	I/O	SDA0
27	P0.4	I/O	N/C
28	P1.25	I/O	N/C
29	P0.5	I/O	N/C
30	P0.6	I/O	N/C
31	P0.7	I/O	BUZZER
32	P1.24	I/O	N/C
33	P0.8	I/O	TXD1
34	P0.9	I/O	RXD1
35	P0.10	I/O	Matrix Key
36	P1.23	I/O	Led8/ LCD (D4)
37	P0.11	I/O	Matrix Key
38	P0.12	I/O	Matrix Key
39	P0.13	I/O	Matrix Key
40	P1.22	I/O	Led7/ LCD (D3)
41	P0.14	I/O	Matrix Key/ EINT1
42	VSS		GND
43	VDD		+3V3
44	P1.21	I/O	Led6/ LCD (D2)
45	P0.15	I/O	Matrix Key
46	P0.16	I/O	Matrix Key
47	P0.17	I/O	Matrix Key
48	P1.20	I/O	Led5/ LCD (D1)
49	VBAT	Input	VBAT
50	VSS		GND

51	VDD		+3V3
52	P1.30	I/O	TMS
53	P0.18	I/O	N/C
54	P0.19	I/O	N/C
55	P0.20	I/O	N/C
56	P1.29	I/O	TCK
57	RESET		
58	P0.23	I/O	P0.23
59	VSSA		GND
60	P1.28	I/O	TDI
61	XTAL2		
62	XTAL1		
63	VREF		+3V3
64	P1.27	I/O	TDO

TECHNICAL SUPPORT

If you are experiencing a problem that is not described in this manual, please contact us. Our phone lines are open from 9:00 AM – 5.00 PM (*Indian Standard Time*) Monday through Saturday excluding holidays. Email can be sent to *support@rhydolabz.com*

LIMITATIONS AND WARRANTEES

This product is intended for personal or lab experimental purpose and in no case should be used where it harmfully effect human and nature. No liability will be accepted by the publisher for any consequence of its use. Use of the product software and or hardware is with the understanding that any outcome whatsoever is at the users own risk. All products are tested for their best performance before shipping, still rhydoLABZ is offering One year Free service warranty (Components cost + Shipping cost will be charged from Customer).

DISCLAIMER

Rhydo Technologies (P) Ltd.
(An ISO 9001:2008 Certified R&D Company)
Golden Plaza, Chitoor Road,
Cochin – 682018, Kerala State, India
Phone : 0091- 484-2370444, 2371666
Cell : 0091- 99466 70444
Fax : 0091 - 484-2370579
E-mail : info@rhydolabz.com, sales@rhydolabz.com
WebSite : http://www.rhydolabz.com

rhydolabz.com